(12) DEMANDE INTERNATIONALE PUBLIÉE EN VERTU DU TRAITÉ DE COOPÉRATION EN MATIÈRE DE BREVETS (PCT)

(19) Organisation Mondiale de la Propriété Intellectuelle
Bureau international

(43) Date de la publication internationale
20 mars 2003 (20.03.2003)

PCT

(10) Numéro de publication internationale
WO 03/024017 A2

(51) Classification internationale des brevets[7] : H04L 9/06

(21) Numéro de la demande internationale :
PCT/FR02/03007

(22) Date de dépôt international :
4 septembre 2002 (04.09.2002)

(25) Langue de dépôt : français

(26) Langue de publication : français

(30) Données relatives à la priorité :
01/11430 4 septembre 2001 (04.09.2001) FR

(71) Déposants *(pour tous les États désignés sauf US)* : **STMICROELECTRONICS S.A.** [FR/FR]; 29, Boulevard Romain Rolland, F-92120 Montrouge (FR). **SAGEM SA** [FR/FR]; 27, rue Leblanc, F-75015 Paris (FR).

(72) Inventeurs; et

(75) Inventeurs/Déposants *(pour US seulement)* : **LIARDET, Pierre-Yvan** [FR/FR]; 56, rue du Pralou, Lotissement L'Audiguier, F-13790 Peynier (FR). **CHABANNE, Hervé** [FR/FR]; 48, Rue de la Marne, F-78200 Mantes la Jolie (FR).

(74) Mandataire : **DE BEAUMONT, Michel**; Cabinet Michel de Beaumont, 1, rue Champollion, F-38000 Grenoble (FR).

(81) États désignés *(national)* : JP, US.

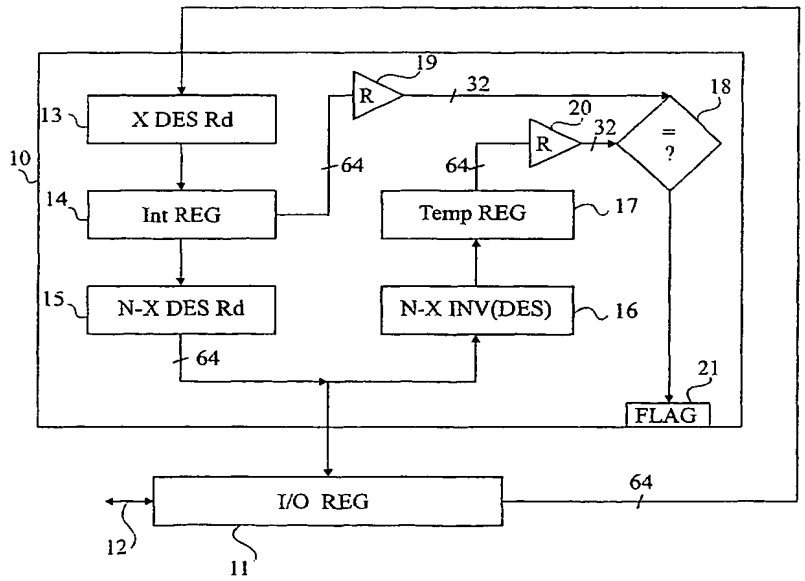
(84) États désignés *(régional)* : brevet européen (AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE, SK, TR).

Publiée :
— *sans rapport de recherche internationale, sera republiée dès réception de ce rapport*

[Suite sur la page suivante]

(54) Title: METHOD FOR MAKING SECURE A SECRET QUANTITY

(54) Titre : PROCEDE DE SECURISATION D'UNE QUANTITE SECRETE



(57) Abstract: The invention concerns a method and a system for making secure a secret quantity, contained in an electronic device, and used at least partly in an encryption algorithm of at least part of an input data executing a predetermined number (N) of successive iterations of a common function and producing at least part of an output data, which consists in: storing (14), after a first number (X) of iterations, an intermediate result; applying, to the output data, a function inverse to that of the encryption for a number (N-X) of iterations corresponding to the difference between the total number of iterations and the first number; comparing (18) the intermediate result with the result of iterations of the inverse function; and validating the encryption only if the two results are identical.

[Suite sur la page suivante]

WO 03/024017 A2

En ce qui concerne les codes à deux lettres et autres abréviations, se référer aux "Notes explicatives relatives aux codes et abréviations" figurant au début de chaque numéro ordinaire de la Gazette du PCT.

(57) Abrégé : L'invention concerne un procédé et un système de sécurisation d'une quantité secrète, contenue dans un dispositif électronique, et utilisée au moins en partie dans un algorithme de chiffrement d'au moins une partie d'une donnée d'entrée exécutant un nombre (N) prédéterminé d'itérations successives d'une même fonction et produisant au moins une partie d'une donnée de sortie, et consistant : à mémoriser (14) , après un premier nombre (X) d'itérations, un résultat intermédiaire ; à appliquer, à la donnée de sortie, une fonction inverse à celle du chiffrement pendant un nombre (N-X) d'itérations correspondant à la différence entre le nombre total d'itérations et le premier nombre ; à comparer (18) le résultat intermédiaire au résultat des itérations de la fonction inverse ; et à ne valider le chiffrement que si lesdits deux résultats sont identiques.

PROCÉDÉ DE SÉCURISATION D'UNE QUANTITÉ SECRÈTE

La présente invention concerne la protection d'une clé ou donnée secrète (généralement un mot binaire) utilisée dans un processus d'authentification ou d'identification d'un dispositif électronique (par exemple, un circuit intégré d'une carte à puce ou une carte électronique contenant un ou plusieurs circuits intégrés) ou analogue, contre des tentatives de piratage. L'invention concerne plus particulièrement la détection d'une tentative de piratage de la donnée secrète, cette détection permettant de bloquer le composant ou le processus utilisant cette donnée secrète, ou encore de simuler un comportement aléatoire.

Parmi les attaques destinées à déterminer par piratage la valeur d'une quantité secrète, l'invention s'applique aux attaques par analyse statistique de fautes (Differential Faults Analysis, DFA) d'un circuit de traitement numérique exploitant une donnée privée ou secrète. Une telle attaque consiste à provoquer une "faute" ou erreur dans l'exécution, par le composant, d'une fonction faisant intervenir une donnée d'entrée (lisible) et la donnée secrète, et à analyser de façon statistique l'influence de cette faute en examinant une donnée de sortie, afin de détecter la donnée secrète. Diverses fautes d'exécution peuvent être provoquées dans le composant. Par exemple, on peut changer la valeur d'un registre interne ou d'un

bit pris en compte dans le calcul, ou encore changer le déroulement du programme interne en le perturbant, par exemple, en accélérant l'horloge d'exécution. On peut encore modifier physiquement le compteur d'instruction, etc. Le plus souvent, lors d'une attaque par analyse statistique de fautes, on perturbe le fonctionnement du composant sans savoir sur quel élément précis on intervient.

Un exemple de système de cryptologie appliqué à une analyse statistique par fautes et un exemple classique de contre-mesure sont décrits dans l'article "Differential Fault Analysis of Secret Key Cryptosystems" de Eli Biham et Adi Shamir paru en 1997 sous les références Technion-Computer Science Department-Technical Report CS0910.revized.

La présente invention s'applique plus particulièrement à la protection d'une clé ou donnée secrète mise en jeu dans un algorithme de cryptographie ou chiffrement d'une donnée d'entrée en exécutant un nombre prédéterminé d'itérations successives d'une même fonction. Par exemple, il s'agit d'un algorithme de type DES (DATA ENCRYPTION STANDARD) décrit, par exemple dans l'ouvrage "Handbook of applied cryptography" de Alfred J. Menezes, Paul C. van Oorschot et Scott A. Vanstone, publié par CRC Press en 1997, pages 252-257. Dans un algorithme DES, une donnée d'entrée est scindée en deux parties (les parties droite et gauche d'un mot binaire) auxquelles on applique par itérations successives une même fonction prenant comme opérandes non seulement la donnée secrète mais également la partie du mot résultant de l'opération précédente, en inversant le côté considéré (droite ou gauche).

La figure 1 illustre, très schématiquement sous forme de blocs, un exemple classique de procédé DES. A chaque itération, on exécute une fonction (bloc 1, F) prenant en compte des parties respectivement droite (R) et gauche (L) d'un mot stocké dans un registre 2. Le résultat de la fonction est ensuite stocké de nouveau dans le registre 2 mais en inversant les positions respectives des parties droite et gauche des mots. Le

nombre d'itérations est variable. En particulier, l'algorithme DES effectue 16 itérations de la fonction F. Afin de rendre le chiffrement et le déchiffrement symétriques, le croisement (inversion des côtés gauche et droit des données résultantes) n'est pas effectué lors de la dernière itération.

Plus généralement, l'invention s'applique à tout algorithme de chiffrement par itérations. Les fonctions mises en oeuvre lors de chaque itération sont souvent des fonctions simples (addition(s), multiplication(s), réduction(s) modulaire(s), permutation(s), substitution(s), etc.) et l'efficacité du chiffrement vient de la répétition de ces fonctions sur les données de sortie de l'itération précédente.

Une attaque par analyse statistique de fautes consiste généralement à intervenir sur la dernière itération d'un algorithme (par exemple, DES). Le plus souvent, on effectue l'opération de chiffrement de la dernière itération, une première fois sans faute et une deuxième fois en ayant provoqué une faute soit dans au moins un bit d'entrée, soit dans l'horloge du programme, soit dans un déroulement quelconque. On combine alors les valeurs obtenues par une addition logique (XOR). En analysant les résultats sur un grand nombre d'opérations, on est en mesure de détecter la quantité secrète mise en jeu. L'erreur volontaire peut être introduite à n'importe quelle itération du calcul. Toutefois, l'analyse des fautes s'effectue toujours sur la dernière itération qui est la seule accessible aux pirates. De plus, dans un algorithme de type DES qui scinde les parties droite et gauche d'un registre, la recherche de la clé s'effectue en examinant uniquement une partie (généralement la partie gauche) des résultats.

Par exemple, on suppose que la dernière itération (16ème) effectue, pour obtenir la partie gauche L_{16} du résultat, l'opération suivante :

$L_{16} = F(R_{15}, K_{16}) \oplus L_{15}$, où F représente la fonction de chiffrage appliquée, où R représente la partie droite du registre résultat (R_{15} représentant son contenu après la 15ème

itération), où L représente la partie gauche du registre résultat (L_{15} représentant son contenu après la 15ème itération), et où K représente la sous-clé mise en oeuvre pour l'itération correspondante (ici, la 16ème).

L'opération effectuée avec une faute provoquée est alors la suivante :

$L_{16}{}^{f} = F(R_{15}{}^{f}, K_{16}) \oplus L_{15}$, où l'exposant f identifie une donnée erronée (entachée d'une erreur provoquée).

Pour la recherche de la clé, on ajoute logiquement les résultats L_{16} et $L_{16}{}^{f}$ et l'on obtient la relation suivante :

$L_{16} \oplus L_{16}{}^{f} = F(R_{15}{}^{f}, K_{16}) \oplus F(R_{15}{}^{f}, K_{16})$, dans laquelle seule la donnée secrète K_{16} est inconnue.

Lors des attaques par introduction de fautes, plus l'erreur est introduite tard dans le processus (sur un résultat intermédiaire de rang élevé), plus le nombre de messages fautifs que l'on doit analyser pour déterminer la clé (plus précisément la sous-clé prise en compte lors de la seizième itération) est réduit. En pratique, on peut considérer que si l'erreur est introduite avant la huitième itération d'un algorithme DES de son itération, le temps nécessaire à la collecte des exécutions entachées d'erreur et à l'exécution automatique de l'analyse statistique devient trop important de sorte que la sous-clé ne peut pas en pratique être piratée. Comme on ne sait pas toujours sur quel rang d'itération on intervient, on utilise fréquemment des attaques aléatoires. Dans ce cas, on a de façon probabiliste, forcément des opérations qui s'effectuent sur les dernières itérations, de sorte que l'on est en mesure de déterminer la sous-clé de façon statistique.

Une première méthode constituant une contre-mesure contre des attaques de type DFA est de dupliquer les calculs. En effectuant deux fois chaque calcul itératif, on considère que l'on est en mesure de détecter si une faute a été introduite lors d'un des calculs. On considère alors qu'il y a peu de risques qu'une même faute se produise deux fois au même moment dans le calcul.

Un inconvénient de cette méthode de contre-mesure est qu'il est nécessaire de reproduire deux fois l'algorithme DES. Si celui-ci est effectué de façon logicielle, cela prend du temps. Si celui-ci est mis en oeuvre de façon matérielle, cela prend de la place par duplication des circuits.

Un autre inconvénient est qu'il est nécessaire de stocker les données finales et intermédiaires dans des registres afin d'être en mesure de comparer les résultats des deux calculs pour détecter une éventuelle attaque.

Un autre inconvénient est qu'il est en fait quand même possible que la même erreur soit reproduite par le pirate avec une probabilité non nulle.

On connaît d'autres procédés de détection de piratage. En particulier, des contre-mesures contre des attaques par analyse statistique de la consommation (Differential Power Analysis, DPA) sont connues de la technique. Ces procédés ne protègent toutefois pas contre des attaques par analyse statistiques d'erreurs (DFA).

L'invention vise à proposer un nouveau procédé de protection d'une donnée secrète contre des attaques par analyse statistique d'erreurs.

L'invention vise plus particulièrement à proposer un procédé de protection qui ne nécessite pas de doubler l'algorithme itératif que l'on souhaite protéger.

L'invention vise également à proposer un procédé particulièrement fiable qui notamment permette d'éviter le risque de voir apparaître deux erreurs consécutives.

L'invention vise en outre à proposer un procédé de protection qui soit peu gourmand, que ce soit en place sur le circuit intégré ou en temps de calcul par rapport à l'algorithme de chiffrement proprement dit.

Pour atteindre ces objets et d'autres, l'invention prévoit un procédé de sécurisation d'une quantité secrète, contenue dans un dispositif électronique, et utilisée au moins en partie dans un algorithme de chiffrement d'au moins une

partie d'une donnée d'entrée exécutant un nombre prédéterminé d'itérations successives d'une même fonction et produisant au moins une partie d'une donnée de sortie, comprenant les étapes suivantes :

mémoriser, après un premier nombre d'itérations, un résultat intermédiaire ;

appliquer, à la donnée de sortie, une fonction inverse à celle du chiffrement pendant un nombre d'itérations correspondant à la différence entre le nombre total d'itérations et le premier nombre ;

comparer le résultat intermédiaire au résultat des itérations de la fonction inverse ; et

à ne valider le chiffrement que si lesdits deux résultats sont compatibles.

Selon un mode de mise en oeuvre de la présente invention, la comparaison s'effectue après application d'une fonction de combinaison et/ou d'une fonction d'expansion et/ou d'une fonction arithmétique, aux résultats intermédiaires.

Selon un mode de mise en oeuvre de la présente invention, la comparaison des résultats intermédiaire et de fonction inverse ne tient compte que d'une partie seulement des données.

Selon un mode de mise en oeuvre de la présente invention, on rend aléatoire l'intervalle de temps entre l'obtention du résultat de l'algorithme de chiffrement et la mise en oeuvre des itérations de la fonction inverse.

Selon un mode de mise en oeuvre de la présente invention, on applique le procédé de sécurisation à la détection d'une tentative de piratage par analyse statistique d'erreurs.

Selon un mode de mise en oeuvre de la présente invention, le nombre d'itérations avant mémorisation du résultat intermédiaire est fonction de la probabilité de découvrir la quantité secrète selon l'itération à laquelle est introduite une erreur.

Selon un mode de mise en oeuvre de la présente invention, le procédé de sécurisation est mis en oeuvre par des moyens matériels.

Selon un mode de mise en oeuvre de la présente invention, le procédé de sécurisation est mis en oeuvre par des moyens logiciels.

Selon un mode de mise en oeuvre de la présente invention, le résultat intermédiaire n'est stocké que pendant la durée nécessaire à sa comparaison avec le résultat issu des itérations de la fonction inverse.

L'invention prévoit également un circuit de chiffrement d'une donnée d'entrée au moyen d'au moins une donnée secrète.

Ces objets, caractéristiques et avantages, ainsi que d'autres de la présente invention seront exposés en détail dans la description suivante de modes de mise en oeuvre et de réalisation particuliers faite à titre non-limitatif en relation avec les figures jointes parmi lesquelles :

la figure 1 décrite précédemment représente, de façon très schématique, une itération d'un procédé DES classique du type auquel s'applique la présente invention ; et

la figure 2 illustre, sous forme de schémas blocs, un mode de mise en oeuvre du procédé de protection de l'invention sous forme matérielle.

Pour des raisons de clarté, seules les étapes de procédé et les constituants d'une cellule de protection qui sont nécessaires à la compréhension de l'invention ont été représentés aux figures et seront décrits par la suite. En particulier, la fonction proprement dite mise en oeuvre par l'algorithme de chiffrement que l'on souhaite protéger n'a pas été détaillée et est quelconque. De plus, les détails du procédé DES auquel s'applique plus particulièrement la présente invention sont parfaitement connus et peuvent être trouvés dans la littérature.

Une caractéristique de la présente invention est de mémoriser, lors de l'exécution du procédé de chiffrement, un résultat de calcul intermédiaire correspondant au résultat de l'algorithme après un nombre d'itérations prédéterminé. Une autre caractéristique de l'invention est, en fin d'algorithme, d'appliquer sur un nombre d'itérations fonction du nombre d'itérations du résultat intermédiaire, une fonction inverse à partir du résultat final. La mémorisation du résultat intermédiaire permet de comparer ce résultat avec celui obtenu lors de l'application des itérations de la fonction inverse. Si ces résultats sont identiques, on peut considérer que le circuit n'a pas fait l'objet d'une tentative de piratage ou que l'erreur provoquée n'est pas exploitable par le pirate.

La figure 2 illustre, sous forme de schéma-blocs, une cellule 10 de chiffrement d'un circuit intégré selon la présente invention. L'exemple de la figure 2 concerne la mise en oeuvre d'un procédé de chiffrement de type DES tel que décrit ci-dessus. On notera toutefois que l'invention s'applique plus généralement à tout algorithme de chiffrement exécutant un nombre prédéterminé d'itérations successives d'une même fonction.

Un message M à chiffrer est, de façon classique, introduit dans un registre d'entrée/sortie 11 (I/O REG) par un bus 12 communiquant avec les autres circuits classiques du circuit intégré (non représentés). Le registre 11 est destiné à contenir, en fin de chiffrement, le message C chiffré. Le nombre de bits des messages M et C dépend de l'application. Par exemple, dans un procédé de type DES, les messages M et C sont généralement sur soixante-quatre bits. Ces soixante-quatre bits du message M sont envoyés en entrée de la cellule de chiffrement 10. Dans l'exemple de la figure 2, on a considéré le cas d'une cellule réalisée par des moyens matériels. En variante, l'algorithme de chiffrement pourra être exclusivement mis en oeuvre de façon logicielle.

En entrée de la cellule de chiffrement, après avoir initialisé, dans un état par défaut, un bit de validation (bloc 21, FLAG) qui sera décrit par la suite, on commence par exécuter un nombre prédéterminé X d'itérations de l'algorithme (blocs 13, X DES Rd). La fonction mise en oeuvre à chaque itération peut correspondre à n'importe quelle fonction d'un algorithme de chiffrement classique. Par exemple, il s'agit de la fonction F d'un algorithme de type DES tel qu'illustré par la figure 1. Le résultat des X itérations correspond au résultat intermédiaire de l'invention, stocké dans un registre dédié (bloc 14, INT REG). Le stockage dans le registre intermédiaire est préférentiellement temporaire, c'est-à-dire que ce registre sera effacé une fois la comparaison effectuée avec le résultat issu de l'application de la fonction inverse comme on le verra par la suite. On termine l'algorithme de chiffrement en exécutant les N-X itérations restantes (bloc 15, N-X DES Rd), où N représente le nombre total d'itérations de l'algorithme de chiffrement (16 pour un algorithme DES). Les soixante-quatre bits résultant de l'application de l'algorithme sont, de façon classique, fournis au registre d'entrée/sortie 11 et correspondent au message C.

Selon l'invention, on applique à ce message, N-X itérations de la fonction inverse de l'algorithme de chiffrement (bloc 16, N-X INV(DES)) de façon à retrouver la valeur intermédiaire stockée dans le registre 14. Le résultat des N-X itérations inverses est stocké dans un deuxième registre temporaire (bloc 17, TEMP REG). Puis, les contenus respectifs des registres 14 et 17 sont comparés (bloc 18, = ?) afin de vérifier qu'ils sont bien identiques. De préférence, la comparaison n'est effectuée que sur une partie des messages contenus dans les registres 14 et 17. En particulier, dans le cadre d'un procédé de type DES, on se contente préférentiellement de comparer la partie droite ou gauche des messages. En effet, en raison des inversions successives des parties droite et gauche à chaque itération de l'algorithme de chiffrement, une telle comparaison est suffisante. Dans ce cas,

les sorties des registres 14 et 17 sur soixante-quatre bits traversent des portes de sélection respectivement 19 et 20 de façon à ne fournir que trente-deux bits au comparateur 18. En variante, les portes 19 et 20 exécutent une fonction quelconque, pourvu qu'elle soit à "collision libre", c'est-à-dire qu'une modification d'un bit d'entrée suffit à modifier la sortie.

Selon un mode de mise en oeuvre préféré de l'invention, la cellule de chiffrement fournit un bit de validation (bloc 21, FLAG) qui, par défaut, est dans un état indicateur d'une erreur (une tentative de piratage). Ce n'est que si le comparateur 18 donne un résultat correspondant à une identité entre les résultats intermédiaire et de fonction inverse (ou une compatibilité entre ces résultats s'ils transitent par une fonction) que le bit de validation 21 commute vers l'autre état. Des résultats sont compatibles si, appliqués à une même fonction (combinaison, calculs des bits de parité, CRC, fonction de hachage, etc.), ils fournissent des résultats égaux. L'état du bit de validation sert, par exemple, à autoriser la fourniture du message contenu dans le registre 11 sur le bus d'entrée/sortie 12. Toute autre utilisation du bit de validation pourra être envisagée. Par exemple, celui-ci peut servir à inhiber d'autres fonctions du circuit intégré tant qu'une authentification n'est pas considérée comme valide. Ou encore, on pourra fournir, en cas de piratage détecté, un résultat aléatoire qui aura pour effet de fausser l'analyse statistique d'erreurs.

Un avantage de la présente invention est qu'elle rend plus difficile le piratage par analyse statistique d'erreurs en rendant plus difficile la reproduction d'une même erreur devant être prise en compte par l'algorithme de chiffrement. En effet, contrairement aux solutions classiques consistant à effectuer deux fois le chiffrement pour lesquelles un pirate éventuel est susceptible de provoquer deux fois la même erreur au même instant dans le déroulement de l'algorithme de chiffrement, une telle reproduction est rendue quasi-impossible par le fait que

la vérification s'effectue sur une fonction inverse. Par conséquent, en provoquant une erreur que ce soit dans les X premières itérations ou dans les N-X itérations restantes de la fonction, une même erreur reproduite au début de la fonction inverse ne conduira pas aux mêmes résultats. Ce résultat conduit à ce que le procédé de l'invention est robuste, même pour des erreurs présentées à des itérations choisies de façon aléatoire.

Selon un mode de mise en oeuvre préféré, l'exécution des N-X itérations de la fonction inverse de l'algorithme de chiffrement est différée avec un délai aléatoire de l'obtention du résultat stocké dans le registre d'entrée/sortie. On rend alors encore moins probable la reproductibilité d'une faute à une même étape de l'algorithme de chiffrement.

Le choix du nombre X d'itérations déterminant le résultat intermédiaire stocké dépend de l'application et de l'algorithme de chiffrement utilisé. Dans l'exemple d'un algorithme de type DES de seize itérations, on choisit préférentiellement de stocker un résultat intermédiaire après huit itérations. Ce choix est lié au fait que, de façon statistique, la clé de cryptage ne peut pas être obtenue par analyse des résultats des huit premières itérations. En effet, si une erreur est introduite pendant les huit premières itérations, l'analyse du résultat du message chiffré ne permettra pas d'obtenir la clé de cryptage en un temps économiquement viable (généralement estimé à quelques mois de collecte de données entachées d'erreurs et de calculs automatiques par ordinateur). Par conséquent, la lecture pirate du registre intermédiaire ne fragilise pas le système. Si l'erreur est introduite entre les neuvième et seizième itérations (bloc 15, figure 2), le pirate éventuel ne parvient pas à reproduire la même erreur au même endroit dans l'application de la fonction inverse sur les itérations 16 à 9 (bloc 16). Cela conduit à ce que le bit de validation (bloc 21) reste dans un état d'erreur.

Dans un algorithme de chiffrement ne prévoyant pas d'inversion ou de mélange des bits des résultats intermédiaires

selon les itérations, la comparaison s'effectuera préférentiellement sur l'ensemble des bits du message afin de ne pas rater la détection d'une erreur si celle-ci est intervenue sur un bit non comparé. Par contre, dans des procédés réalisant une inversion de parties des messages à chaque itération comme c'est le cas pour l'algorithme DES, on peut se contenter de ne comparer qu'une partie des messages. En effet, la probabilité de ne pas détecter une attaque par l'introduction d'une erreur est alors négligeable et on gagne un temps considérable sur l'opération de comparaison.

Bien entendu, la présente invention est susceptible de diverses variantes et modifications qui apparaîtront à l'homme de l'art. En particulier, on pourra choisir ou non d'effectuer un certain nombre d'opérations en parallèle. Par exemple, si l'algorithme de chiffrement est mis en oeuvre de façon matérielle, on peut utiliser les temps de lecture/écriture dans les registres pour effectuer en parallèle certains calculs notamment certaines itérations de la fonction inverse de l'algorithme de chiffrement.

De plus, la réalisation pratique de l'invention et son adaptation à un algorithme de chiffrement classique par itérations successives est à la portée de l'homme du métier à partir des indications fonctionnelles données ci-dessus que ce soit pour une mise en oeuvre logicielle ou matérielle. La fonction F et les inversions de la figure 1 correspondent, dans cet exemple, à une des N itérations.

En outre, l'invention s'applique que la donnée secrète soit utilisée en tout ou en partie dans chaque itération.

Enfin, le procédé de l'invention est compatible avec les procédés classiques constituants des contre-mesures aux attaques par analyse statistique de la consommation.

REVENDICATIONS

1. Procédé de sécurisation d'une quantité secrète, contenue dans un dispositif électronique, et utilisée au moins en partie dans un algorithme de chiffrement d'au moins une partie d'une donnée d'entrée exécutant un nombre (N) prédéterminé d'itérations successives d'une même fonction et produisant au moins une partie d'une donnée de sortie, caractérisé en ce qu'il comprend les étapes suivantes :

mémoriser (14), après un premier nombre (X) d'itérations, un résultat intermédiaire ;

appliquer, à la donnée de sortie, une fonction inverse à celle du chiffrement pendant un nombre (N-X) d'itérations correspondant à la différence entre le nombre total d'itérations et le premier nombre ;

comparer (18) le résultat intermédiaire au résultat des itérations de la fonction inverse ; et

ne valider le chiffrement que si lesdits deux résultats sont compatibles.

2. Procédé selon la revendication 1, caractérisé en ce que la comparaison s'effectue après application d'une fonction de combinaison et/ou d'une fonction d'expansion et/ou d'une fonction arithmétique, aux résultats intermédiaires.

3. Procédé selon la revendication 1 ou 2, caractérisé en ce que la comparaison des résultats intermédiaire et de fonction inverse ne tient compte que d'une partie seulement des données.

4. Procédé selon l'une quelconque des revendications 1 à 3, caractérisé en ce qu'il consiste à rendre aléatoire l'intervalle de temps entre l'obtention du résultat de l'algorithme de chiffrement et la mise en oeuvre des itérations de la fonction inverse.

5. Procédé selon l'une quelconque des revendications 1 à 4, caractérisé en ce qu'il est appliqué à la détection d'une tentative de piratage par analyse statistique d'erreurs.

6. Procédé selon la revendication 5, caractérisé en ce que le nombre d'itérations avant mémorisation du résultat intermédiaire est fonction de la probabilité de découvrir la quantité secrète selon l'itération à laquelle est introduite une erreur.

7. Procédé selon l'une quelconque des revendications 1 à 6, caractérisé en ce qu'il est mis en oeuvre par des moyens matériels.

8. Procédé selon l'une quelconque des revendications 1 à 6, caractérisé en ce qu'il est mis en oeuvre par des moyens logiciels.

9. Procédé selon l'une quelconque des revendications 1 à 8, caractérisé en ce que le résultat intermédiaire n'est stocké que pendant la durée nécessaire à sa comparaison avec le résultat issu des itérations de la fonction inverse.

10. Circuit de chiffrement d'une donnée d'entrée au moyen d'au moins une donnée secrète, caractérisé en ce qu'il comporte des moyens pour mettre en oeuvre le procédé de sécurisation selon l'une quelconque des revendications 1 à 9.

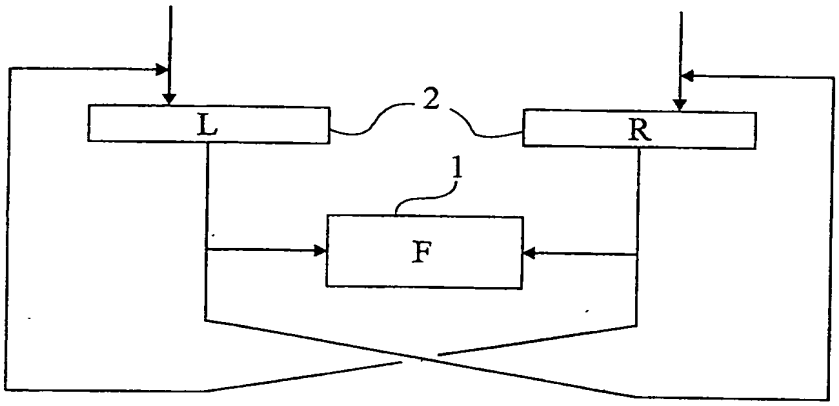


Fig 1

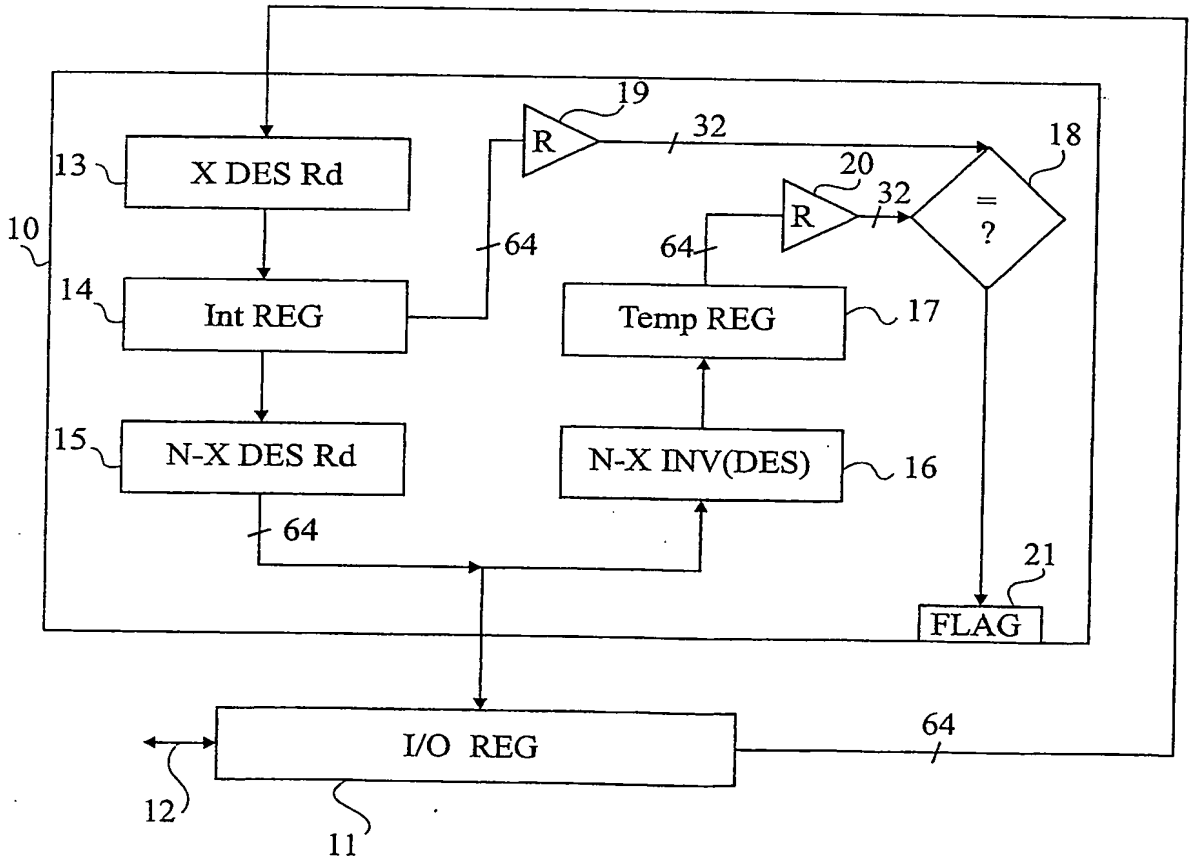


Fig 2